Vision-based terrain learning

Robert E. Karlsen[a] and Gary Witus[b]
[a]U.S. Army –TARDEC, Warren, MI 48397-5000
[b]Turing Associates, Ann Arbor, MI 48103

ABSTRACT

This paper presents an algorithm for online image-based terrain classification that mimics a human supervisor's segmentation and classification of training images into "Go" and "NoGo" regions. The algorithm identifies a set of image chips (or exemplars) in the training images that span the range of terrain appearance. It then uses the exemplars to segment novel images and assign a Go/NoGo classification. System parameters adapt to new inputs, providing a mechanism for learning. System performance is compared to that obtained via offline fuzzy c-means clustering and support vector machine classification.

Keywords: terrain classification, computer vision, machine learning, exemplar memory

1. INTRODUCTION

Monocular and stereo video cameras continue to be the most practical vision sensor for small inexpensive robots. Unfortunately, unstructured vision-based navigation continues to be an especially difficult problem. In this paper, we present an approach to automated image segmentation and terrain classification using exemplars, or small image samples, to represent the variety of terrain appearance.

Exemplars are used as cluster seeds to segment the terrain. Local pieces of terrain are assigned to the exemplar to which they are most similar in appearance. The pieces of terrain then inherit the terrain class membership of the exemplar. Exemplar models assume that intact stimuli are stored in memory, and that classification or recognition is determined by the degree of similarity between a stimulus and the stored exemplars. Simple generalization effects explain correct classification of novel (previously unseen) instances of categories. Only the item information is used for classification decisions. Categorization relies on the comparison of a new stimulus with known exemplars of the category.

Exemplar models are the most parsimonious models of categorization in terms of the underlying associative mechanism[1]. Exemplar based learning was originally proposed as a model of human learning in Ref. [2], and has since been shown to explain both human and animal visual classification performance significantly better than alternative hypotheses of feature-based and prototype-based processing.[3,4]

Various researchers have begun to develop methods to forecast traversability using estimates of geometrical properties inferred from non-contract sensors. References [5] and [6] developed a fuzzy-rule-based system to mimic human "high/medium/low" trafficability assessment based on measures of roughness, slope and distance between obstacles computed from stereo imagery. The system was targeted for planetary rover environments. Reference [7] used a stereo color vision system together with a single axis LADAR to classify terrestrial terrain cover and detect obstacles. They noted that the color-based classification system could be made more robust by considering the texture of regions and the shape features of objects. Reference [8] defined a trafficability index equal to the weighted sum of the slope and roughness, estimated from line-scanning laser rangefinder data. Reference [9] classified terrain as impassible (NoGo) if any of several properties were above a threshold: height variation, the surface normal orientation, and the presence of an elevation discontinuity (all estimated from LADAR imagery). Reference [10] developed a rule-based system for terrain classification from LADAR and color camera imagery.

Appearance based approaches do not attempt to directly estimate geometrical properties and then infer traversability. Instead, they associate the operator's assessment of trafficability directly from the terrain appearance. The operator's

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **17 APR 2006**	2. REPORT TYPE **Journal Article**	3. DATES COVERED **17-04-2006 to 17-04-2006**

4. TITLE AND SUBTITLE **VISION-BASED TERRAIN LEARNING**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Robert Karlsen; Gary Witus**	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Army TARDEC ,6501 E.11 Mile Rd,Warren,MI,48397-5000**	8. PERFORMING ORGANIZATION REPORT NUMBER **#15683**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **U.S. Army TARDEC, 6501 E.11 Mile Rd, Warren, MI, 48397-5000**	10. SPONSOR/MONITOR'S ACRONYM(S) **TARDEC**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) **#15683**

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
2006 SPIE DEFENSE AND SECURITY SYMPOSIUM

14. ABSTRACT
N/A

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

trafficability assessment is not restricted to geometrical properties, but can also reflect surface properties (e.g., friction, resistance, sinkage) and factors that do not affect traversability, but which nonetheless exclude certain terrain (e.g., the risk of being run over by a car or the need to avoid detection by staying in shaded areas).

Fig. 1: Input training image and classification.

Various applications could benefit from automatic methods to segment and classify terrain from images, such as virtual reality simulated terrain, mobile robot navigation, combat engineering planning, and land cover analysis for ecological studies. These applications address different scales, terrain features and classes of interest. It is unlikely that any specific segmentation and classification criteria would be suitable for all of these applications. Nonetheless, the applications have important similarities. In all cases, we implicitly assume that local areas with similar appearance should be grouped together in any segmentation, and that they are likely to be representatives of the same terrain class. We also implicitly assume that we know in advance what terrain classes we are interested in and what they commonly look like. For the purposes of this research, we assume that the segmented terrain regions or regions of the same terrain class do not have any a priori constraints on their geometric shape or global organization. We also assume that there are no a priori constraints regarding which terrain classes can be adjacent to each other.

The approach is currently implemented as a software system designed to provide considerable flexibility in the choices of perspective transformation, resolution, scale, sampling and difference metric. In general, different choices will be appropriate for different applications. The software automatically builds a characteristic "basis set" of exemplars from training images. We currently build a set of exemplars for each terrain class, with the union over the terrain classes being the basis set exemplars for an application. A second option is to build a set of terrain segmentation exemplars independent of the terrain classes, and then associate the exemplars with terrain classes. In its present form, the algorithm does not attempt to resolve ambiguities when an area does not resemble any of the a priori terrain classes, or areas that have partial membership in two or more terrain classes. Instead, it produces a fuzzy classification, i.e., a segment of terrain can have partial membership in different terrain classes, and may be partially unclassified.

2. TECHNICAL APPROACH

The algorithm is organized into two routines: one for training and one to apply segmentation and classification. At the end of training, the exemplar bank and associated data are stored in a file to be loaded before applying the segmentation and classification.

2.1 Training images and overlays

The user must provide a set of representative training images. Ideally, the training images would be drawn from the same distribution as the downstream application images. In practice, this may not be possible. The effect on segmentation and classification performance of different terrain, foliage, season, lighting, and weather between the training image set and test/application image set is a question for empirical investigation. In principle, the images can be multi-spectral with an arbitrary number of planes. The current algorithm requires that the images be RGB or monochrome images stored in a standard image format.

For each training image, a corresponding terrain classification overlay is required. The overlay denotes which locations correspond to which terrain class. One approach is to use an N plane image, where N is the number of terrain classes and each plane is a binary image. An alternative approach is to use a single plane image, using integer values from 1 to N (for the N terrain classes), and zero for unclassified locations. This representation is more appropriate when there are a large number of terrain classes, or when the terrain classes constitute an ordered set, e.g., ordered by traverse ability cost or by speed-made-good. For purposes of demonstration, we use two terrain classes (e.g., "Go" and "NoGo" regions) and the overlays are stored as three-plane RGB images (the third plane is not used). The terrain classification is displayed as an RGB image in which one terrain class is coded red and the other is coded green, with blue used to code unclassified regions. An example of this is shown in Fig. 1, where the gravel driveway is designated as a "Go" region and everything else is designated a "NoGo" region.

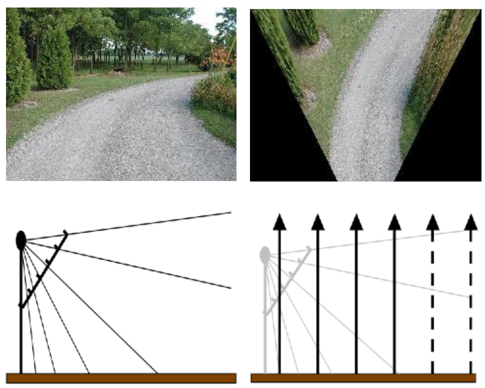

Fig 2: Camera image view and pseudo plan view.

2.2 Perspective transformation, resolution, scale and sampling

In some cases, a transformation from original camera perspective may be appropriate. In the camera image view, pixels represent the same angle (assuming lens distortion effects are minimal), but do not project onto equal areas of ground. Assuming the elevation of the camera is large relative to the variation in ground elevation in the scene, the pseudo plan view projection can be used to create a new image in which each pixel corresponds to the same ground area (see Fig. 2). The pseudo plan view projection is good for areas where the variation in elevation is small relative to the elevation of the camera, but produces distortion when this is not the case. An alternative projection is to restrict analysis to horizontal sub-bands within the image. The band view does not distort vertical objects, but retains the perspective distortion of the original camera image for flat earth regions.

Both the pseudo plan view and camera view options are supported in the current software. Both transformations require the size of the camera image, and the angle subtended by an individual pixel (we assume square pixels). The pseudo plan view projection requires three additional inputs: (1) the height of the camera above ground plane, (2) the distance on the ground from the spot below the camera to the ground projection of the bottom row of the image, and (3) the desired resolution of the projected image, i.e. the pixel width of the output projection in centimeters.

The camera band view also requires three additional inputs: (1) the image row number of the top row of the band, (2) the image row number of the bottom row of the band, and (3) the resolution for the band-view image (the angle of pixels in the band view image must be less than or equal to the pixel angle of the original camera image).

The user must also specify the analysis scale for terrain segmentation and classification. The segmentation and classification is based on exemplar image chips (square chips in the current software). The scale is the width of the exemplar chips. Membership in a terrain class is considered to be a bulk property of a local region, not a point-location property. The user must also specify the center-to-center spacing, or sampling distance, for the output segmentation and classification images.

2.3 Image space transformation

The purpose of the image space transformation is to amplify the importance of selected image properties. For example, the imagery can be transformed into a variety of color spaces. The importance of color could be strengthened or weakened by weighting different image planes. In addition to the RGB color coordinate system, we have experimented with the HSV (hue, saturation, value) and L*a*b* (luminance, red/green, yellow/blue) systems.

Another transformation option is to adjust the high spatial frequency content relative to low spatial frequency content by constructing a multi-resolution pyramid representation and then applying weights to the image planes. A common example is the laplacian-of-gaussian spatial bandpass pre-filtering often used in stereo-vision processing.

The space transformation could increase the dimensionality of the image space. Consider a monocular image input. The image could be processed through a bank of N spatial filters, such as edge and corner filters at different spatial scales and orientations. Each filter produces a single-plane output image.

2.4 The exemplar basis set

The current algorithm processes the training images one at a time. The current image is chopped into chips at the specified scale and sampling distance. There is an option to find exemplars of each image independent of exemplars from other images, or to find only new exemplars sufficiently different from exemplars built from preceding images. If the former option is selected, all chips are nominated as potential exemplars. If the exemplar processing is in the context of previous exemplars, only chips whose minimum distance (in terms of the image metric) to existing exemplars is

greater than the current clustering threshold are nominated as potential exemplars, i.e., chips that resemble current exemplars are not considered as possible new exemplars.

Each chip is compared to its neighbors within a specified radius to calculate the difference metric between it and each of its neighbors (the radius is a user input). The aggregate local difference between the chip and its neighbors is calculated as the weighted average of the mean and minimum differences (The weight is a user input. Weighting towards the minimum leads to a larger pool of exemplars, and weighting towards the mean leads to a smaller pool of exemplars). Chips similar to their neighbors are preferred over those that are different.

The algorithm calculates a clustering threshold equal to the weighted sum of the minimum and maximum local differences over all chips (The weight is a user input. Weighting towards the minimum leads to a larger pool of exemplars and tighter clusters. Weighting towards the maximum leads to a smaller pool of exemplars and broader clusters). This threshold provides the system's adaptation ability. Training images with significant variability provide coarser segmentation over training images with lower variability, for the same size of exemplar bank.

Exemplars for the current image are selected iteratively. Initially, no chips are rejected. Of the non-rejected chips, the one with the minimum local difference is added to the bank of exemplars. All chips, whose difference from the exemplar is less than the clustering threshold, are rejected. This process is iterated until all chips have either been added to the exemplar bank or rejected. The exemplars for the current image are then merged with the bank of exemplars from the previous images.

2.5 Image chip difference metric

Image difference metrics remain an open issue in the evaluation of image compression schemes. While it is easy to measure the amount of compression and the encoding/decoding time, it is not clear how to measure the quality of the reconstructed image, i.e., its difference in appearance from the original. Different image characteristics are important depending on the image content, the questions at hand, and who is looking at the image.

Similarly, there is no obviously correct metric for measuring the difference between two images. Before the images are chopped into chips, they can be processed to balance the relevant image characteristics (see II.C Image Space Transformation). In principle, therefore, simple measures of the aggregate difference are all that are needed. Even so, there are many different ways to calculate the difference between two image chips. Some metrics are computed from the pixel-by-pixel difference between two chips, others are computed from the difference in statistics of the individual chips, e.g.,

- the sum over all pixel locations and all image planes of the absolute value of the difference between the two images;
- the root sum square over all pixel locations and all image planes of the difference between the two images;
- the maximum over all image planes of the sum over all pixel locations of the absolute value of the difference between the two images;
- the sum over all pixel locations of the maximum over all image planes of the absolute value of the difference between the two images;
- the root sum square over all image planes of the difference in the mean values and difference in standard deviations (over pixel locations) of the two images; and
- the sum over all image planes of the absolute difference in the mean values and difference in standard deviations (over pixel locations) of the two images.

The first four metrics are computed from pixel-by-pixel differences of the image chips, while the last two metrics are computed from statistics of the image chips. Although, the software is set up to incorporate different metrics, the results in this paper are based on the first and last metrics.

2.6 Exemplar membership in terrain classes

Each image chip maps to a region in the terrain classification overlay. The terrain classification of the image chip is simply the expected membership in each of the terrain classes. It is possible that a chip could straddle more than one terrain class, or could straddle an unclassified portion of the overlay. After the new exemplars are added to the exemplar bank, the current image is segmented using all of the exemplars in the bank. Each chip location in the image is assigned

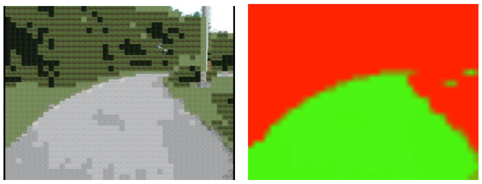

Fig 3: Reconstruction of training image from exemplars and resulting classification.

to the exemplar to which it is closest, provided the distance is less than the current clustering threshold. In some cases, some image chips may not be associated with any exemplar. For each exemplar in the bank, we accumulate the number of times the exemplar is "hit" by an image. The terrain class membership of the exemplar is the mean over all chips associated with the exemplar, of terrain class memberships of the chips. The terrain segmentation is converted to terrain classification by assigning each location the terrain class membership values of the exemplar associated with that image location.

2.7 Output illustration controls

The algorithm contains options to output different images to illustrate and provide insight into the processing:

- the pseudo plan view or camera band view perspective transformation of the image;
- the pseudo plan view or camera band view perspective transformation of the terrain class overlay;
- the exemplar chips (at their location in the image) selected from the current image;
- the segmentation of the current image based on the current bank of exemplars; and
- the classification of the image based on the current bank of exemplars.

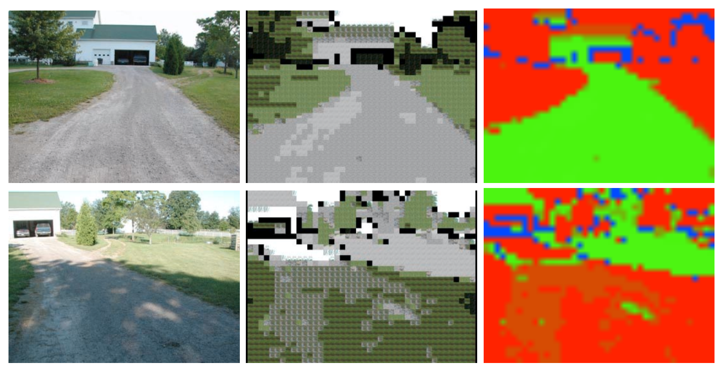

Fig. 4: Test images, reconstruction from exemplars, and resulting classification. (One RGB training image)

There is no single best way to represent the different segments for purposes of visualization. Color-coding shows the different segments, but does not give much insight into the basis for the segmentation. The software illustrates the segmentation in a way that provides direct visual insight into the basis for the segmentation. To visualize the segmentation, the software replaces each image chip with the exemplar chip that it is associated with (image chips not associated with any exemplar appear black and the classification is coded with blue) (See Figs. 3 and 4). When the sampling distance is less than the exemplar scale, the exemplars are blended in the reconstruction. The visualization image is the same size as the pseudo plan view or camera band view perspective image, so it is easy to directly compare the two. By using the exemplar chips themselves, the visualization image shows what the exemplars look like, and which image chips they are associated with. Finally, comparing the visualization to the perspective image gives prima fascia evidence of the credibility of the segmentation.

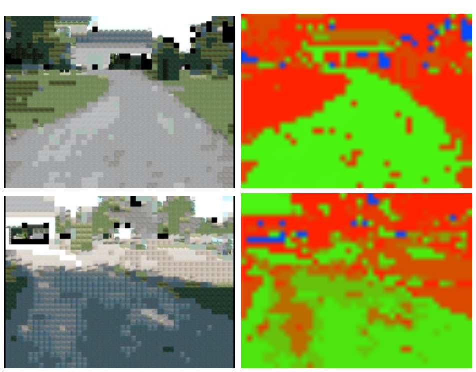

Fig. 5: Reconstruction from exemplars and resulting classification. (Two RGB training images)

2.8 Application for segmentation and classification

The application routine reads in the filter bank and associated data produced by the training routine. It segments and classifies the test images one at a time. No changes are made to the exemplar bank or associated data. After pseudo plan view or camera band view perspective processing, the test image is chopped into chips at the specified scale and sampling distance. Each image chip is assigned to the closest matching exemplar, providing the match is within the current clustering threshold, otherwise the chip is unassigned. This produces the segmentation by exemplars. After the segmentation, each location is assigned the terrain class fuzzy membership of the segmenting exemplar. The classification image is at the resolution of the center-to-center sampling distance.

3. DEMONSTRATION RESULTS

This section illustrates the segmentation and classification system. The demonstration uses color-coding to show the terrain classification into "Go" (green), "NoGo" (red), and "Unclassified" (blue) regions. Fig. 4 shows classification results derived from the single training image in Fig. 1, where gravel is designated "Go" and everything else is "NoGo." The image data consisted of the three RGB color planes and the distance metric was computed from the pixel-by-pixel chip difference. This training resulted in 25 exemplars. Note the errors due to the building in the upper image and in the lower image due to the shadowed gravel. Adding a second training image similar to the lower image in Fig. 4, results in the classification results of Fig. 5, with 78 exemplars. Note the overall improvement in the shadowed region and in the grassy areas. However, the upper image classification has become noisier.

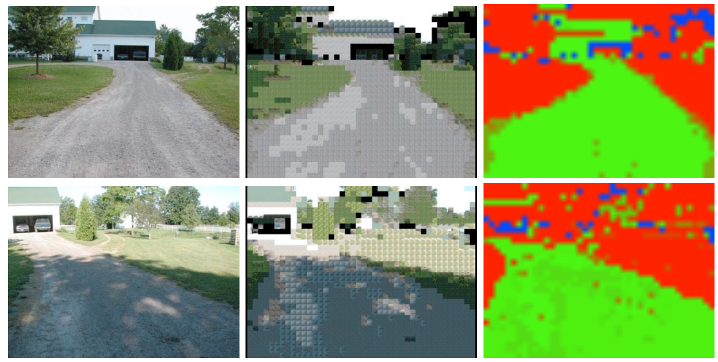

Fig. 6: Test images, reconstruction from exemplars, and resulting classification. (Two L*a*b* training images)

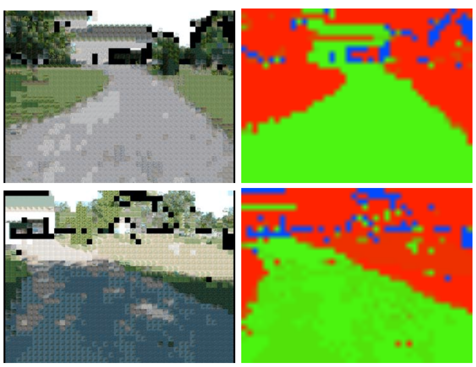

Fig. 7: Reconstruction from exemplars and resulting classification. (Two L*a*b* training images with texture)

To compensate for different lighting conditions, we implemented a conversion to the HSV (hue, saturation, value) color space, in order to separate the color information from the luminance. Although this resulted in some improvements, at the expense of more exemplars, the HSV system is unsatisfactory due to the cyclical nature of hue and the fact that HSV is far from perceptually uniform. This led to the use of the L*a*b* color space transform, where L^* refers to luminance and the a^* and b^* components encode the color information (red/green and yellow/blue differences, respectively). The transformation to L*a*b* is nonlinear, resulting in components that are nearer to perceptually uniform.

Figure 6 shows the results of training the algorithm with images transformed to the L*a*b* color space. The upper image is similar to the RGB classification, while the lower image is much improved. However, the number of exemplars has increased by a factor of two to 172. Note that the images in Fig. 6 are from the original RGB color space, not the L*a*b* color space, as the latter are more difficult to interpret visually.

Color alone is not always a good indication of image matching, and therefore we have also included texture as an additional dimension on which to differentiate and compare image exemplars. Figure 7 shows the results of adding a texture plane, computed by taking the standard deviation of a sliding window throughout the image. The classification is smoother, but not significantly better than without texture on these two images and the number of exemplars increased to 278.

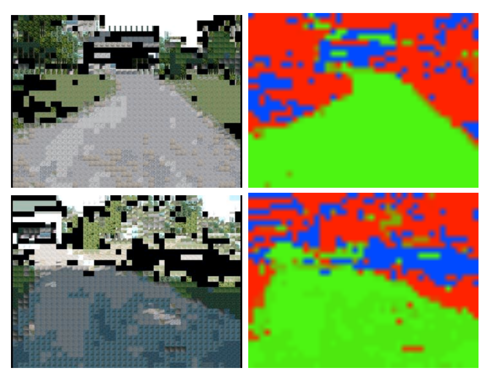

Fig. 8: Reconstruction from exemplars, and resulting classification. (Two L*a*b* training images with texture and using statistical differences)

All the preceding analysis was performed using a difference metric based on computing pixel-by-pixel differences between the image chips. There is also the option of computing statistics on each image chip and then computing the difference between the statistics. Figure 8 shows the results of using a distance metric that is the sum of the absolute differences of the mean and standard deviation over each image plane. This segmentation required only 75 exemplars, similar

to the number for the previous RGB classification with no texture in Figure 5, but with improved classification accuracy. Although there are more unclassified segments, in most cases these would be coded "NoGo" for cautious driving. The low number of features when using statistical measures makes the use of other learning algorithms such as neural networks, support vector machines, or clustering, more feasible. Memory requirements are also reduced, since only the statistics of the exemplar are stored, not the entire chip.

Fig. 9: Input training images and classification for extended comparison.

4. COMPARISON TO OTHER TECHNIQUES

To compare our online classification methodology to other techniques, we turned to a more realistic and difficult problem, using the same set of images. Instead of segmenting out gravel from everything else, we segmented out safe driving areas, which tended to be gravel and grass for this data set, which consists of two similar image sequences. Figure 9 shows the two training images, which are the same as for the preceding analysis, and their associated segmentation masks. We chose 23 other images from the two image sequences to test the algorithms, which required hand drawing the classification maps for each of the test images. Because there were 1344 feature vectors for each image, the training set consisted of 2688 samples and the test set had 30,912 samples.

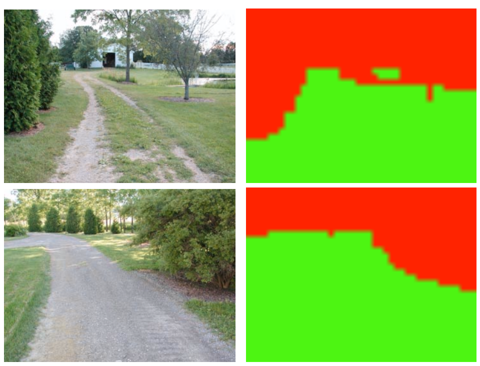

Fig. 10: Test images and hand-drawn classification maps.

4.1 Fuzzy c-means clustering

Since the online algorithm produces exemplars that are essentially cluster centers, it is natural to compare the performance against a standard clustering algorithm, such as fuzzy c-means clustering (FCM).[11] Because the online difference metric uses the absolute difference between feature vectors, while the FCM algorithm computes a root-mean-square difference, we took the square root of the feature vectors before passing them to the FCM algorithm. We also replaced the cluster centers, as computed by the FCM algorithm, with the closest feature vector, in order to replicate the use of exemplars and to compute the reconstruction images. The online algorithm analyzes each class separately, and then compares the test feature vectors against exemplars from each class. We replicated this behavior in the FCM algorithm by partitioning the two classes in the training data and computing clusters for each separately. We modified the code from Ref. [12] for our implementation of the FCM algorithm.

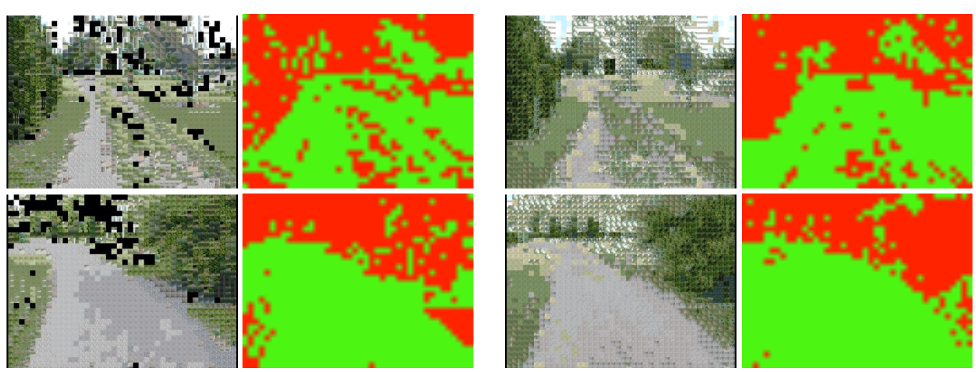

Fig. 11: Test image reconstruction from exemplars and resulting classification for the online (left) and clustering (right) methods. (Two L*a*b* training images with texture)

One issue with typical clustering algorithms is the requirement to choose the number of clusters beforehand. There are a number of validation measures that can be used to select an optimum number of clusters, but we have not explored them sufficiently to determine if any, in fact, correlate with classification accuracy. Instead, we

computed the test error as a function of the number of clusters and chose a cluster number where the test error flattened out. This is not acceptable, in general, unless the validation set is separate from the testing set, which was not the case with this data set. We are currently exploring a validation method involving the redundancies seen when cluster centers are replaced by exemplars. For the current case, we chose 20 clusters for each class, for a total of 40 clusters.

We also implemented a metric to compare the output classification mask to a user-drawn mask. Fully “Go” regions are mapped to +1, fully “NoGo” regions are mapped to -1, and unknown regions are mapped to 0. The classification accuracy metric is the absolute difference between chips divided by 2. While this metric is appropriate for the fuzzy classification in the previous section, in order to compare different methods, we defuzzified the classification map and mapped the unknown regions to “NoGo,” which would normally be done in any case for cautious driving.

Figures 11 show the clustering results for the two test images of Figure 10. Note that while the clustering algorithm tends to produce more accurate classification maps, the differences are not overly large. This is borne out by the average classification accuracy for the two methods, where the combined classification error over the 23 test images is 0.163 with 92 exemplars for the online method, while the FCM algorithm had an error of 0.115 with 39 exemplars. Choosing 10 clusters per class for a total of 20 exemplars for the FCM algorithm, results in an error of 0.137, whereas choosing 40 clusters per class, leading to 73 exemplars, gives an error of 0.138.

4.2 Support vector machines

We also compared the clustering algorithms with the classification results from a support vector machine (SVM) analysis. While there are published SVM algorithms for clustering analysis[13], we used a previously developed classification implementation. Therefore, the results are meant to provide a comparison to the previous algorithms, rather than as a substitute for either of them.

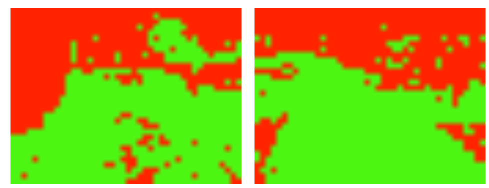

Fig. 12: Classification results from the SVM algorithm. (Two L*a*b* training images with texture)

The SVM algorithm is a wide margin classifier that finds a set of parallel hyperplanes separating the data, such that the perpendicular distance between the hyperplanes is maximized. A kernel function can be used to transform the data into a higher-dimensional space, such that when the separating hyperplanes are transformed back into the original space, they become curved surfaces. Standard SVM algorithms include an adjustable parameter to handle non-separable data, allowing the user to set the importance of excluding data in the margin between the separating hyperplanes.[14,15]

Figure 12 shows classification results for the same images as in Figure 10, where the SVM training resulted in 602 support vectors. The kernel function selected was a quadratic polynomial that we have found gives good results for sparse data sets, such as we have used here. Since the SVM algorithm was not a clustering algorithm we do not show reconstruction images. The overall classification accuracy was 0.109, which is only slightly better than the FCM results, providing support for the performance of the FCM clustering.

It appears that a good choice for a complete system would be a combination of the FCM clustering algorithm and the online learning algorithm demonstrated here. The FCM clustering algorithm could be used for the initial offline training, while the online learning algorithm would be employed when the system is performing its mission. The use of both the FCM and SVM algorithms will allow us to explore different variations of the features that we have examined here, as well as providing a baseline for other types of features that we may consider adding.

5. FINDINGS AND OBSERVATIONS

This paper has demonstrated an online approach to image-based terrain segmentation and classification using exemplars. Exemplars provide a simple way to represent the characteristic color/luminance and spatial patterns of terrain. Since the exemplars are drawn from training images in such a way as to span the appearance of the training images, they are well suited to represent the variations of appearance without an a priori model of terrain appearance. The software system, as presented, allows for considerable flexibility in specifying the perspective transformation, image space transformation,

scale, resolution, sampling density, and image difference metric. Empirical research is needed to tune these options for specific applications.

Preliminary results indicate the approach has potential to segment terrain in a manner that is consistent with subjective perception. The segmentation appears to be robust over changes in lighting, specific terrain, and automatic camera gain and contrast adjustments. Our preliminary results indicated that analysis in the camera band view was more useful for segmenting and classifying positive obstacles than the pseudo plan view. When presented with novel images, the camera band view was more likely to produce mixed Go/NoGo terrain classification, whereas the pseudo plan view was more likely to produce unclassified terrain segments. This may be due to the fact that the camera band view mixes different scales, whereas the pseudo plan view maintains more consistent scale.

With only a limited training set, the online algorithm still performed quite well on the simplistic segmentation of gravel from other terrain. When presented with a combination of both grass and gravel, the system still performed reasonably well. Nonetheless, the preliminary analysis is not adequate to assess the value of this method of terrain classification for any specific application, e.g., robot navigation. More extensive testing, with a structured experimental objectives and design are needed to evaluate the applicability of this method of terrain classification for any specific application. The algorithm is reasonably fast, with the largest time consumption actually being the reconstruction of the segmentation images by inserting exemplars. But this step is for visualization purposes only. The results presented here do not take advantage of the additional information provided by fuzzy classification.

The online algorithm results compared favorably to the results obtained from offline fuzzy c-means (FCM) clustering. Although the latter performed measurably better, it had the advantage of seeing all the data at once. The online algorithm is image order dependent and is therefore suboptimal. The data was partitioned and transformed in order to make the comparison as close as possible. One difference that was not addressed is that the FCM algorithm identifies each test chip with a cluster, whereas the online algorithm marks chips whose distance from any exemplar is above a set threshold as “Unclassified.” On the other hand, the online algorithm does use information about neighboring image chips in making decisions. The latter information could also potentially be used with the FCM clustering algorithm.

If we continue to use the FCM clustering algorithm as an offline training method, we will need to find a method for selecting the number of clusters. We will analyze the standard set of validation measures to determine if they correlate with classification accuracy. We will also explore two other potential validation measures: one involves the redundancy seen when substituting exemplars for cluster centers and the other involves using the support vector machine (SVM) algorithm to find a lower threshold for the training classification error. The baseline FCM algorithm finds spherical clusters of the same size, and so we will also examine the efficacy of algorithms that provide more general cluster shapes and sizes.

Additional future work involves training and testing on a larger set of images, as well as applying the algorithm to video streams and implementing on a mobile robot. Since terrain appearance varies as a function of distance, fusing range data from a stereo camera system, with the color and texture information currently being used, is anticipated to provide enhanced performance. We also plan to explore other types of texture measures, such as those provided by filtering with various structure elements, such as lines, corners, or other shapes. The architecture we have set up allows these additional features to be simply added as additional image planes.

REFERENCES

1. S. Chase and E.G. Heinemann, “Exemplar memory and discrimination,” in *Avian Visual Cognition*, R.G. Cook, Ed., 2001.
2. D. Medin and M. Schaffer, “Context theory of classification learning”, *Psychological Review* **85**(3), 207-238, 1978.
3. R.M. Nosofsky, “Tests of an exemplar model for relating perceptual classification and recognition memory,” *Journal of Experimental Psychology: Human Perceptual Performance* **17**(1), 3-27, 1991.
4. C.W. Werner and G. Rehkamper, “Categorization of multidimensional geometrical figures by chickens (Gallus gaullus f. domestica): fit of basic assumptions from exemplar, feature and prototype theory,” *Animal Cognition* **4**, 37-48, 2001.

5. A. Howard, E. Tunstel, D. Edwards, and A. Carlson, "Enhancing fuzzy robot navigation systems by mimicking human visual perception of natural terrain traversability," Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf., Vancouver, B.C., Canada, pp. 7-12, July 2001.
6. A. Howard, and H. Seraji, "Vision-based terrain characterization and traversability assessment," *J. Robotic Systems* **18**(10), pp. 577-587, 2001.
7. R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots* **18**, 81-102, 2005.
8. C. Ye and J. Borenstein, "A method for mobile robot navigation on rough terrain," Proceedings of the IEEE International Conference on Robotics and Automation, 3863-3869, 2004.
9. D. Langer, J.K. Rosenblatt, and M. Hebert, "A behavior-based system for off-road navigation," *IEEE Trans. On Robotics and Automation* **10** (6), 776-783, 1994.
10. A. Sarwal, D. Simon, and V. Rajagopalan, "Terrain classification," Proc. SPIE Vol. 5083, *Unmanned Ground Vehicle Technology V*, 156-163, 2003.
11. F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, John Wiley & Sons, 1999.
12. B. Balasko, J. Abonyi and B. Feil, "Fuzzy clustering and data analysis toolbox" (www.fmt.vein.hu/softcomp/fclusttoolbox/).
13. A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research* **2**, 125-137, 2001.
14. V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, 1998.
15. C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **2**, 1-47, 1998.